6. Python: de basis

De structuur van een Python-programma

In vorig hoofdstuk heb je een eerste Python programma gezien Dit geeft een idee van de opbouw van de programma's. Een programma bestaat uit een aantal **instructies** of opdrachten. In het voorbeeld staat elke opdracht op een aparte regel.

Naamgeving

Namen duiden variabelen, functies, klassen of andere objecten van Python aan. Een naam begint met een letter (a..z of A..Z) of een underscore (_), gevolgd door letters en/of cijfers. Leestekens of speciale tekens mag je niet gebruiken. Namen zijn hoofdlettergevoelig. Import, import en IMPORT … zijn verschillende namen en kunnen een verschillende betekenis hebben.

De meeste programmeurs laten de naam van een *klasse* (zie later) beginnen met een hoofdletter, de andere namen meteen kleine letter. Namen die met een underscore beginnen zijn meestal private variabelen. (zie later)

Volgende woorden zijn **gereserveerd.** Die mag je niet als naam gebruiken.

and	elif	if	print
as	else	import	raise
assert	except	in	return
break	exec	is	try
class	finally	lambda	while
continue	for	not	with
def	from	or	yield
del	global	pass	

Commentaar

Met **commentaar** kan je je programma verduidelijken. Dat is belangrijk als andere programmeurs jouw programma gebruiken. Commentaar begint met teken #. Alles achter dit teken tot het einde van de regel is commentaar en wordt genegeerd bij het uitvoeren van het programma. Wil je een stuk commentaar over meerdere regels? Zet dan aan het begin en het einde drie enkele aanhalingstekens.

Voorbeeld:

```
# ons eerste programma

c = 5
k = 7
```

```
# we tellen 2 getallen op
d = c+k   # het resultaat is d
’’’
Dit is
Commentaar van 2 regels
’’’
```

Inspringen

Opdrachtregels die bij elkaar horen laten we inspringen. Voorbeeld:

```
c = 5
if (c>3):
    print ("c>3")
    a = c
    print (a)
```

De drie regels na de if-opdracht horen bij elkaar. Als c groter is dan 3, dan worden die drie regels uitgevoerd.

Opdrachten van meerdere regels en lege regels

De backslash (\) geeft aan dat de opdracht verder gaat op de volgende regel.

```
Oppervlakte = lengte * \
breedte
```

betekent

```
Oppervlakte = lengte * breedte
```

Binnen de (), [], of {}-haken is de backslash is niet nodig.

De interpreter negeert lege regels.

Aanhalingstekens

Met aanhalingstekens noteren we strings, stukken tekst. Je mag enkele of dubbele gebruiken, maar gebruik ze wel consequent: vooraan en achteraan dezelfde! We gebruiken drie dubbele aanhalingstekens achter elkaar om een string over meerdere regels te splitsen.

```
Naam = ‘Jan Mertens’
Gemeente = “Schoten”
Tekst = “””Dit is een lange
tekst over meerdere regels.”””
```

Een programma opstarten

In vorig hoofdstuk hebben we een eerste programma gemaakt en uitgevoerd met de startknop van de werkbalk van Thonny. Bij het opstarten vraagt Thonny waar het moet opgeslagen worden: This computer of *MicroPython Device*. Welke plaats je kiest, je kunt altijd een bewaard programma terug oproepen in Thonny om het op te starten. Wil je dat een programma automatisch opstart na het opstarten van de controller of na een reset, dan moet je het bewaren op de controller met naam **main.py**.